AR-008-614 (2)

GRAFTED - USER MANUAL

M.E. BUCKLAND, M. WEBSTER AND FRANK TKALCEVIC

MRL-GD-0057

DECEMBER 1993

MATERIALS RESEARCH LABORATORY

DSTO

GRAFTED - GRAphical Fault Tree EDitor: A Fault Tree Description Program for Target Vulnerability/Survivability Analysis - User Manual

Accesion For	
NTIS CRA&I	
DTIC TAB	
Unannounced	
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and / or Special
A-1	

Michael E. Buckland, Mark Webster and Frank J. Tkalcevic

MRL General Document
MRL-GD-0057

Abstract

A computer program GRAFTED, "GRAphical Fault Tree EDitor", has been written to simplify data entry and modification of component fault tree descrip ns (FTD) used in military platform vulnerability/survivability analysis. GRAFTED uses a unique, graphical, screen based data entry procedure to define and display both individual system component parameters, and their hierarchical relationship in the overall system FTD. The generated component and system FTD output is in a format that is directly readable by the MRL version of the General Vulnerability Assessment Model. (GVAM), suite of computer programs.

Although GRAFTED was specifically designed to generate FTDs for GVAM, it could be easily modified to accommodate data input formats and FTD output for other assessment procedures that require user friendly data entry and graphical fault tree editing and visualisation.

This version of GRAFTED was written specifically for Tektronix UNIX workstations, and only runs on these machines.

DEPARTMENT OF DEFENCE
DSTO MATERIALS RESEARCH LABORATORY

Published by

DSTO Materials Research Laboratory
Cordite Avenue, Maribyrnong
Victoria, 3032 Australia

Telephone: (03) 246 8111
Fax: (03) 246 8999

AR No. 008-614

Contents

Contents (Contd)

GRAFTED - GRAphical Fault Tree EDitor: A Fault Tree Description Program for Target Vulnerability/Survivability Analysis - User Manual

1. Grafted and Fault Trees

1.1 Introduction

This manual details the use of GRAFTED, a fault tree editor that was originally conceived to support work involved in assessing combat system survivability, with the data required for use by the MRL vulnerability codes. However the program output data can be easily manipulated for other uses requiring fault tree analysis.

The program provides ease of input of complicated system layouts. The GRAFTED data files can be easily changed to allow for modifications due to updated systems, or for isolating possible problem systems.

GRAFTED allows systems to be described as a collection of individual components, which are linked to produce a tree structure. Each component is individually labelled, and information relevant for further processing is entered via a user friendly interface.

To accommodate intricate systems which may rely on a number of support systems, the tree is made up of smaller modules. This allows for rapid modification of the tree structure, because modules can be moved and altered at any time without the entire tree being re-entered.

Damage algorithms for the various failure mechanisms involved for individual components can give a reliability rating, or probability of failure for various components. However, the relative importance and effectiveness of these individual components to the overall performance of a system cannot always be easily assessed, especially for large, multi-component systems. Such systems may have a large number of interconnections between components. The overall assessment of system failure should be simple, and reliable.

1.2 *The Fault Tree*

The fault tree is a component layout of a system to allow for failure analysis exercises to be carried out. GRAFTED has a user friendly interface that allows easy entry of component information relevant to component failure analysis.
The fault tree output of GRAFTED is in a format that allows easy manipulation by other programs. This allows the information to be easily assessed.

1.3 *Basic Layout*

The fault tree developed by GRAFTED is broken down into a series of interconnecting components. The component description will depend on the system, and type of analysis being performed (ie the detail involved in the system layout depends on the study involved). The basic format of the tree is as follows (see figure 1):

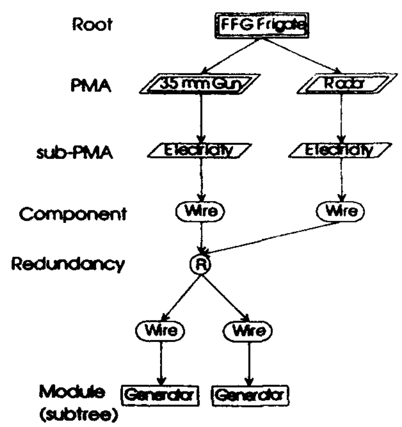

Figure 1: Basic tree layout

The tree structure is developed from the top down, with the top of the tree being referred to as the Root Node. The Root Node is the name of the platform being analysed and gives a reference point to start the fault tree.

1.4 *PMA - Primary Mission Areas*

The platform being analysed can be divided into several independent systems, or Primary Mission Areas (PMAs) which can share common components, but perform independent functions in the overall analysis of the platform. The overall effect of these independent systems to the performance of the overall platform can depend on the mission being performed by the platform.

Examples of PMAs that could be used in an analysis of a weapons platform include:

MOB - mobility
AAW - anti-air warfare
ASW - anti-submarine warfare
ASUW- anti-surface unit warfare

1.5 Sub - PMAs

Each PMA being analysed can be divided into one or more independent functions that contribute to that PMA; these functions are identified as Sub–PMAs. The relative effectiveness that each function contributes to the overall effectiveness of the PMA is input at this stage and is used in the calculation of the survivabilty of the PMA.

The actual component fault tree starts at the sub–PMA level and is a series of interconnecting components, represented by nodes making up a description of the system layout.

1.6 Module Reference

For ease of developing intricate fault trees, which may have numerous component inter–connections and components shared between a number of functions, the tree is developed as a series of Modules or blocks of components that are self contained. This allows for tree structures to be easily modified without the whole tree being regenerated.

Each module, or block of components, is saved in an individual file. This allows GRAFTED to call individual modules into memory to be edited. This means that the whole tree is not in memory, and therefore it is less likely for accidental losses or changes to occur. A module is placed in the fault tree as shown in the example, figure 1. At completion of an editing session GRAFTED will ask for the file to be recompiled and the entire tree is re–configured and saved to a file called *<name>.fta*. This file gives a complete description of the fault tree layout.

1.7 Components

The basic unit of the fault tree is a component. The components are interconnected to form the tree (see figure 1). Each node represents a component. The size and detail that a layout contains are dependent on the analysis required. The component data that are required for failure analysis are input directly and this information is saved as a separate file *<name>.cmp*. This file gives a description of all the components within the tree.

1.8 Redundancy

Redundant branches are included in the tree by use of a special node (the redundancy node). The relative effectiveness of each branch to the overall effectiveness of the function is input at this stage. (Please read Fault Tree Analysis section to ensure appropriate use of this node)

1.9 Fault Tree Analysis

GRAFTED allows for the input and editing of the fault tree, which is used in the overall assessment of survival probability. To ensure that the data are input correctly it is necessary to understand the fault tree analysis procedure.

The calculation of the probability of survival for a given model is as follows:

1. Physical damage models are used to assess the kill probability for each component, $pk_i(c)$ using the component data contained in the file <name>.cmp. The probability of survival for each component $ps_i(c)$ is therefore :

$$ps_i(c) = 1 - pk_i(c)$$

2. Using the fault tree description, the probability of survival for a block of components or a module is

$$ps_j(m) = \prod_{i=1}^{n} ps_i(c) = \prod_{i=1}^{n} [1 - pk_i(c)]$$

2b. Survivability at a redundant node is calculated as follows:

$$ps_j(r) = 1 - \prod_{i=1}^{n} \left[1 - \frac{eff_i \cdot ps_i(c)}{\text{maxeff}} \right]$$

where: eff_i = effectiveness of Sub-PMA
maxeff = maximum effectiveness value (should be 1.00)

In most circumstances the effectiveness values for each branch in the redundant node will be 1.00 (each branch equal value), that is, if one branch is disabled the other branch can fully replace the function of this branch. In certain circumstances one branch is not fully effective in replacing the damaged branch, and the effectiveness value used in this case is the relative effectiveness of an individual branch compared to the most effective branch (which is assumed to be able to perform the full function).

It is not appropriate to use the Redundancy Node for situations in which branches perform a certain percentage of the total function of a module. In this case, the tree structure must be manipulated somehow to arrive at an appropriate configuration.

3. The probability of survival for each PMA is calculated as follows:

$$ps_i(PMA) = \sum_{i=1}^{n} eff_i \cdot ps_i(SPMA)$$

where: eff = effectiveness of Sub-PMA

and

$$\sum_{i=1}^{n} eff_i = 1$$

4. A similar equation to that above can be used to calculate the overall survivability of a platform for a given mission with more then one PMA; however, the effectiveness of each PMA in different scenarios may change. Therefore this calculation can be performed outside the program using the values for PMA survivabilities that have been calculated.

2. *Using Grafted*

2.1 *Creating a New Fault Tree Description*

To create a new fault tree in a given sub-directory, run the creation program by typing:

```
new_fault_tree
```

at the UNIX prompt, in the appropriate sub-directory. This program will create all the relevant files required to begin a new fault tree description.

NOTE - The program GRAFTED requires each fault tree description to exist in its own sub-directory. If an attempt is made to create a second tree in a sub-directory, the new_fault_tree program will stop with an error message like this:

```
A fault tree description may already exist in this directory!
Please delete the relevant files before continuing:
LAST_COMPONENT_ID LAST_MODULE_ID MODULE_LIST
NODE_LIST
```

This indicates that a fault tree may already exist in this directory. To overwrite the tree, the listed files must first be deleted. The files used by GRAFTED are described in section 2.8, 'GRAFTED Files'.

2.2 *Starting the Program*

The program is run by typing:

```
grafted
```

at the UNIX prompt, which is the name of the executable program. The program expects a fault tree description to be present in the current directory; if there isn't, the computer will respond with this message:

> A fault tree model does not exist in this directory.
> Change to the directory of the fault tree, or to create
> a new tree, create a new directory and run "new_fault_tree"

If GRAFTED does find a current fault tree description in the present working directory (pwd), the program will load, and present an empty tree editing screen as shown in figure 2.

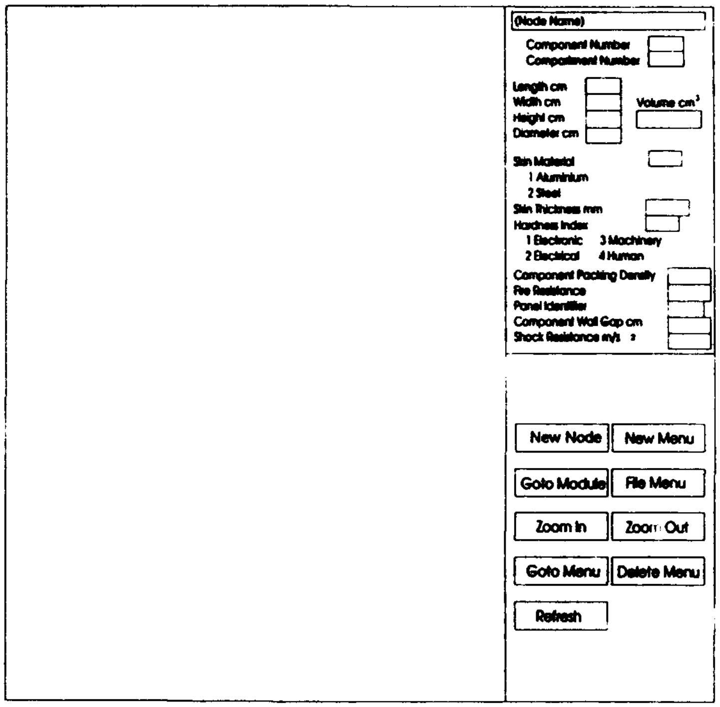

Figure 2: GRAFTED's main screen

The large screen area is the Tree Editing Area, where the graphical representation of the Fault Tree Description will appear. The upper-right section is the Node Data Area, where all the data relevant to the selected node will appear. The lower-right section is the Command Area, which contains the buttons that operate the command list boxes.

2.3 *Using List Boxes*

List boxes are used to provide an easy way to select an item from a list. They are activated by selecting one of the command menus. Figure 3 below shows two example list boxes.

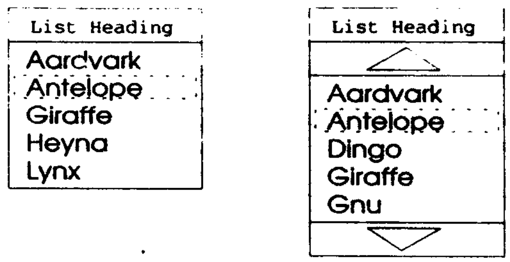

Figure 3: Examples of list boxes.

The list box can appear in two forms. The list box on the left contains only a small number of items in the list and can therefore be drawn in a smaller box. The box on the right has more data items in the list than could fit in a small box. The scroll arrows indicate there are more data to be viewed.

2.3 1 *Using the keyboard with a list box*

The up and down arrow keys are used to move the cursor up and down one entry at a time. The 'Prev Screen' and 'Next Screen' keys are used to move the list of data one boxful at a time. The 'Return' key is used to select the highlighted entry, while the 'Esc' key is used to cancel the list box.

The cursor can also be moved through the list of items by typing the name of the object. For example, typing the letter *g* will move the cursor in the second box to the 'Giraffe' entry. Following this with the letter *n* will move the cursor to the 'Gnu' entry. A small text cursor will also move from left to right as you type the name. Hitting the 'backspace' key will move the text cursor position back one. Hitting any of the other scroll keys will move the small cursor to the far left.

2.3.2 *Using the mouse with a list box*

Clicking on an entry in the list box with the mouse will select the entry. Clicking on the scroll arrows will scroll the list box text one page at a time. To cancel the list box, click the mouse on any other part of the screen.

2.4 *Building a fault tree*

There are two modes in GRAFTED: Tree Editing Mode, and Data Editing Mode. Tree Editing Mode allows new nodes to be added to the fault tree, or nodes to be removed. It also allows the connections between nodes to be edited. This takes place in the Tree Editing Area.

The Data Editing Mode allows the data attached to a given node in the fault tree to be edited; this takes place in the Node Data Area.

To toggle between these modes, simply click on the appropriate area (ie. the Tree Editing Area or the Node Data Area). Hitting the ESC key while in Data Editing Mode will also return to Tree Editing Mode.

To create a new fault tree ensure the current mode is Tree Editing Mode, and select the New Node option by clicking once on the New Node button (or by using the Insert key). GRAFTED will display a rectangle with a double border. This is the Root Node. The Root Node has no data fields attached to it other than a name. It simply acts as a base from which the tree can grow—hence the name Root Node.

The next level down a fault tree from the Root Node is the PMA level (see section 1.4). GRAFTED represents PMAs with double bordered parallelograms. To add a PMA use the Insert key or the New Node Menu as before. To add more PMAs select the root node again by clicking on it or by using the up arrow, and insert another PMA.

The fault tree can continue to grow down through the levels described in section 1 by selecting a node and adding another node underneath. A full example of creating a fault tree is given below in section 3.

2.5 *Editing Component Data*

To edit the data belonging to the various nodes, select the appropriate node, and then enter Data Editing Mode. This will cause a cursor to appear in the node name text area. The text areas in the Node Data Area toggle between overstrike and insert mode upon pressing the Insert key (if the current mode is overstrike the letters OVR will appear in the bottom, right hand corner of the screen). The Return key will advance the cursor through the text areas in the Node Data Area; or by using the mouse, any of the text areas can be activated directly.

After the required alterations are made, return to Tree Editing Mode.

2.6 *Editing Effectiveness Data*

The Edit Effectiveness Button will appear above the menu buttons when the selected node is a PMA. Clicking on this button will display a list box containing the names of all the relevant Sub-PMAs, with a default effectiveness of 1.00.

To edit the effectiveness select the appropriate Sub-PMA from the list box, press the Return key, and type in the new value. Pressing the Return key again will accept this new value.

Clicking in the Tree Editing Area or hitting the ESC key will return to Tree Editing Mode.

2.7 *Grafted Hot Keys*

All the functions that can be executed with the mouse in GRAFTED can also be invoked using the keyboard. This allows for more efficient use of the program. Below is the list of keys that can be used to replace or supplement the action of the mouse.

Key Combination	GRAFTED Function
Return	if a node is selected, Edit Node Data if a module is selected, Goto Module (selected Module)
Remove	Delete Node
Insert	if in Tree Editing mode, Add Node if in Data Editing Mode, toggle insert/overstrike*
Find	Goto Module Menu
Prev Screen	Goto root node of current Module
Next Screen	Goto last node of current Module
← ↑ ↓ →	Scroll through tree
Ctrl-R	Refresh Screen
Ctrl-N	New Menu
Ctrl-F	File Menu
Ctrl-G	Goto Menu
Ctrl-D	Delete Menu
Ctrl-I	Zoom In
Ctrl-O	Zoom Out
Ctrl-B	Back to Previous Module
Ctrl-E	Edit Effectiveness for Redundancy nodes and PMA nodes

* When overstrike mode is active the letters OVR appear in the bottom-right corner of the screen.

2.8 *Grafted Files*

GRAFTED uses a large number of files, all which can be found in the directory of the fault tree model. The files and their use are shown below:

LAST_COMPONENT_ID	This file contains an integer corresponding to the last component identifier used for this tree description.
LAST_MODULE_ID	This file contains an integer corresponding to the last module identifier used for this tree description.
MODULE_LIST	This data file contains a list of all modules used in the tree description.
NODE_LIST	A machine readable data file which contains data on all the data nodes in the fault tree description.
MOD?????	A series of data files starting with the letter "MOD" and ending in a five digit number above 20 000 inclusive. These files contain the data for each module defined in the fault tree.

NOTE: The files used by GRAFTED are in a complicated format and should *not* be modified manually.

3. Developing a Fault Tree - The Generic Missile

3.1 Introduction

To illustrate the range of options available to the modeller a 'Generic Missile', which is representative of any modern guided missile, is used as an example of a fault tree model. The Generic Missile serves as an excellent example because it contains all the tree components provided by GRAFTED.

3.2 The Generic Missile Description

The Generic Missile is a self guided, medium range, air breathing, subsonic, anti-surface missile (see figure 4).

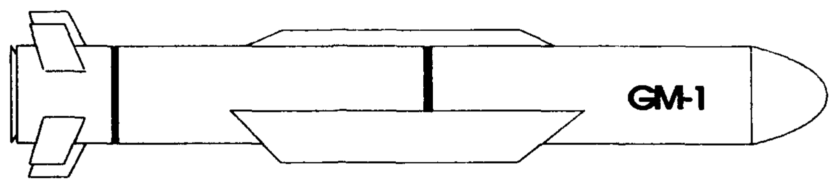

Figure 4: The Generic Missile.

3.2.1 Dimensions

Length:	4.0 m
Diameter	35.0 cm
Wing span	1.0 m

3.2.2 Propulsion

The Generic Missile is propelled by a small turbofan jet engine. There is an air intake on the under side of the main missile body. Fuel is supplied from the fuel tank by an electrical fuel pump.

3.2.3 Guidance

The Generic Missile relies on Inertial Navigation and Active Radar Homing for guidance to the intended target. It has a main radar unit, a radar altimeter, and a gyroscope unit. A small digital computer calculates the navigational data from the radar, radar altimeter, and gyroscope inputs. These data are interpreted by the Actuator Control Unit which engages the Control Fin Actuators. The two actuators are each connected to two opposing Control Fins that steer the missile (see figure 5).

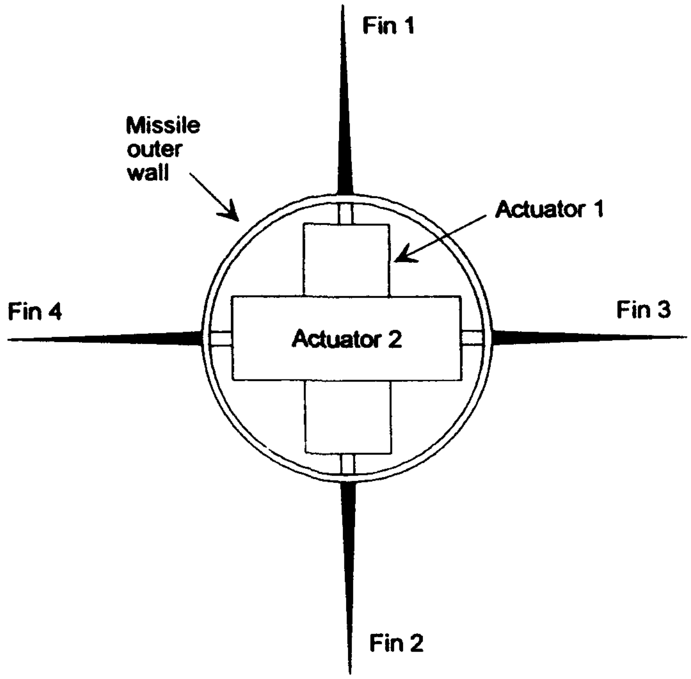

Figure 5: Cut away diagram showing the fin actuators in the Generic Missile.

3.2.4 *Flight*

The Generic Missile also has two wings to provide aerodynamic lift.

3.2.5 *Warhead*

The Generic Missile has a blast/fragmentation warhead with a safe/arm contact fuse. Upon impact with a solid object the fuse will initiate a booster which in turn detonates the high explosive filling.

3.2.6 *Electrical Power*

The Generic Missile is powered by a dry chemical battery.

3.3 *Modelling the Generic Missile with GRAFTED*

Once a basic knowledge of the proposed model (in this case the Generic Missile) and how it functions is established, GRAFTED may be used to build up a fault tree model.

3.3.1 Root Node

When GRAFTED starts, the first task is to add a Root node, and to name it. Use the 'Insert Here' key, or New Node button as mentioned in section 2.4. The root node is represented by a double bordered rectangle. Enter Data Editing Mode, and type "Generic Missile" in the Node Name text area (Section 2.5). Remember to continue blanking out old data with the space bar if necessary in overstrike mode. Hit the Return key when satisfied the name is correct. To return to Tree Editing Mode hit ESC or click anywhere in the Main Tree Graphics Area.

3.3.2 Primary Mission Areas

It must now be decided how many PMAs to include for the Generic Missile.

Any ordnance has one basic purpose - destroying (or damaging) a target. For the Generic Missile to fulfil this purpose it must propel itself towards the target, find the target and steer to intercept it, and detonate upon hitting the target. If any of these three functions fails to some degree, the Generic Missile's success will be limited. Therefore it may be appropriate to choose three PMAs based on these operational requirements: Navigation, Flight, and Warhead.

To add these PMAs, ensure Tree Editing Mode is active, and add three nodes under the Root node. Note, the Root node will have to be re-selected after each addition because, by default, the newest node is the current selection. GRAFTED represents PMAs with double bordered parallelograms.

PMAs have two data fields: a name, and the relative effectivenesses of their Sub-PMA 'children' (see section 2.6). To edit the name of a PMA ensure it is selected then enter the Data Editing Mode as before. It must now be decided how to split up each PMA into Sub-PMAs.

3.3.3 Sub-PMAs

Consider the Navigation PMA. The missile must be able to do two things to successfully navigate its way to the target: locate the target, and steer towards it; so it may be appropriate to choose Guidance, and Control, as the two Navigation Sub-PMAs. These nodes are added and named as above.

To edit the effectiveness of these Sub-PMAs ensure that the Navigation PMA is selected, and follow the procedure in section 2.6. Because the missile would be equally useless without the ability to find the target or to steer towards it, it may be appropriate to choose to leave the effectivenesses of these Sub-PMAs set to 1.00.

The ability to maintain aerodynamic flight relies on maintained propulsion, and structural integrity of the wings and fuselage. So the Flight PMA may similarly be split into two Sub-PMAs: Airframe, and Propulsion. These are added as above and, once again, because the missile would be useless without either of these abilities, the effectiveness of each may remain at 1.00.

The Warhead function is relatively straight forward, and may not require splitting into Sub-PMAs. Simply adding a single Sub-PMA named Warhead may be sufficient.

The Fault Tree Description is now complete down to the Sub-PMA level and should resemble figure 6.

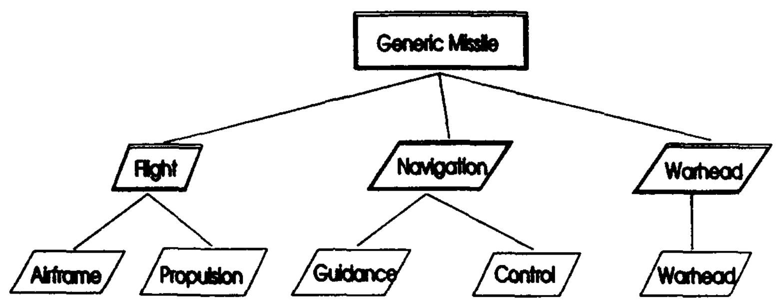

Figure 6: Fault tree Description for Generic Missile down to Sub-PMA level.

3.3.4 Components

Now most of the logical structure of the model is in place the components have to be added to their appropriate Sub-PMAs. Consider the guidance Sub-PMA. The Generic missile relies on both Inertial Guidance and active radar homing. Therefore the likely components to include in this section of the fault tree are a radar dish, a radar unit, a computer, a gyroscope unit, and a radar altimeter (Section 3.2.3).

To add these components, ensure the Guidance Sub-PMA is selected and add five nodes (note that as a node is added, it becomes the current selection). Each of these nodes may be edited in turn to complete their description in the Node Data Area.

Similarly, the Control Sub-PMA will require the components: Control Fins (x4), Control Fin Actuator, and the computer. The Control Fins present an opportunity to illustrate the Redundancy feature of GRAFTED; the missile will retain some manoeuvrability as long as some of the fins remain intact.

From Figure 5 it is seen that damaging Fin 1 or 2 will remove yaw control; damaging Fin 3 or 4 will remove pitch control. For simplicity, the assumption used here is that losing either pitch or yaw control reduces the effectiveness of the control Sub-PMA by 50%. To represent this in the fault tree a redundancy is added after the Control Fin Actuator and Computer. To do this, click on the New menu, and choose Add Redundancy from the list box. This will add a Redundancy node, with no 'children', to the tree. Nodes can now be added to the Redundancy node as normal. For this instance, two nodes must be added. This requires re-selecting the Redundancy node after adding the first 'child'. Each of the nodes now below the Redundancy node must also have a node added under them. The new nodes can now be re-named and the Fault Tree Description should now resemble figure 7.

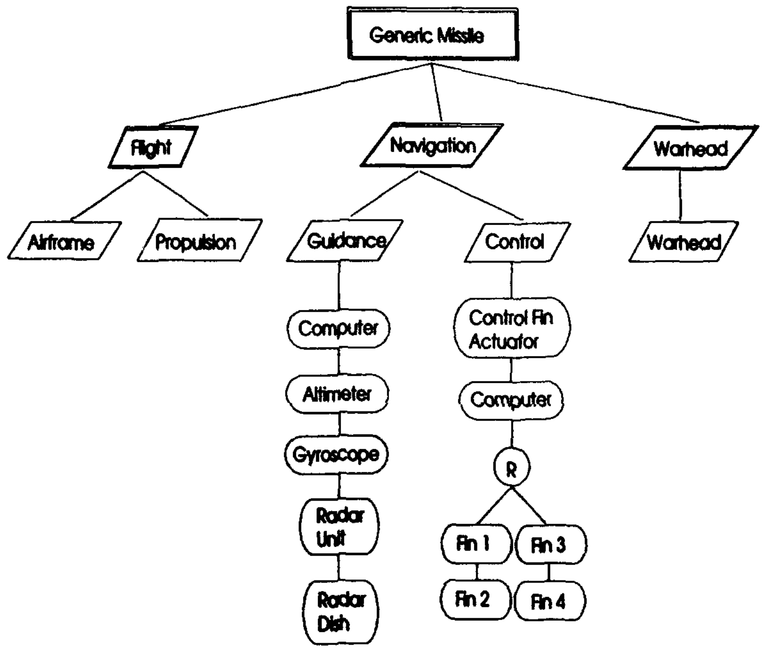

Figure 7: The Fault Tree Description for The Generic Missile Including the Navigation Components.

For the sake of brevity, the addition of the other components of the Generic Missile to their respective Sub-PMAs will not be discussed. The procedures are identical to those for the Navigation PMA described above.

3.3.5 *Modules*

The Sub-PMAs shown in figure 7 are not quite complete. All the components shown require electrical power to operate. This provides an opportunity to illustrate the use of modules.

Rather than repeating a given set of components throughout the fault tree, GRAFTED allows for the construction of modules which may contain several components, including Redundancies. In other words, Modules are like mini, self-contained fault trees.

To create a new module, first close the current module (which is the Root module) by clicking on the File menu, and selecting 'Close Module' from the list box. This will clear the Tree Editing Area. Inserting a new node now will create a new module, similarly to how the Root node is created when the program is first run (see section 2.4).

The module may now be renamed and components added in exactly the same way as before. An appropriate name may be 'Power'. A battery and some cable may be added, and the Power module should resemble figure 8.

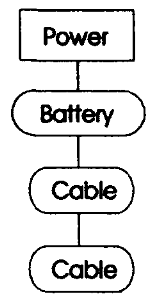

Figure 8: Example of a Power Module.

This module may now be saved, and connected to any point in the main tree that may be deemed appropriate. To save the module, click on the File menu, and choose 'Save Module'. To return to the main tree, select the root module (which in this case is the only other module) by clicking on the Goto Module menu.

To add the Power Module to the main tree, select the appropriate node (in this case the Radar Dish) click on the New menu, and select Add Module. This will cause a list box of available modules to appear; only the Power Module will appear in this case. Clicking on the chosen module will add that module to the tree under the currently selected node.

To add the Power Module to the bottom of the Control Sub-PMA, first add the module to one of the fins, Fin 2 for example. Then Select Fin 4, and add the Power Module to it also. The Fault tree will now resemble figure 9.

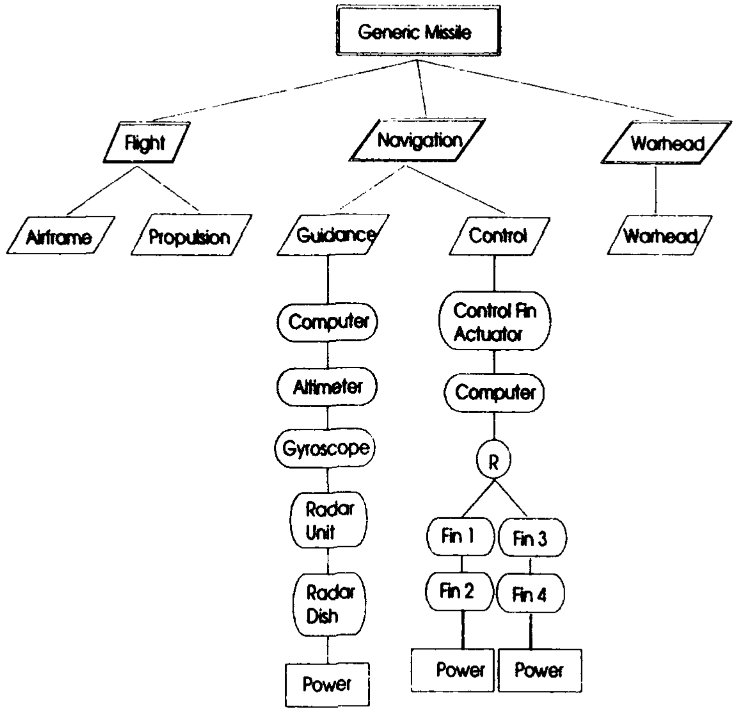

Figure 9: Fault Tree Description of Generic Missile With Power Module.

3.4 Summary

The remainder of the Generic Missile model will not be described in full, but all the procedures necessary for modelling the missile completely have been discussed. Adding nodes including: the Root Node, PMAs, Sub-PMAs, components, redundancies, and modules have all been described, and examples shown.

Upon exiting GRAFTED (Exit under the File Menu) the program will ask if the changes should be saved, and the Fault Tree Description files re-compiled. The next time GRAFTED is run from this directory, the Generic Missile model will be loaded automatically but not displayed. To display a given Module, use the Goto Module command under the Goto Menu.

4. GRAFTED Command Reference

4.1 New Menu

The New Menu is selected to create a new object in the fault tree description. The menu can be invoked by selecting the New icon with the mouse, or hitting Ctrl-N. The menu options are shown below

New Menu
Add Node
Add Redundancy
Add Module
Add Root
Add Branch

4.1.1 Add Node

The Add Node function will create a new node in the fault tree. It can also be executed by hitting the "INSERT HERE" key, or selecting the "New Node" icon. The type of node created is dependent on the parent node. The following table shows the type of node created with a given parent.

Parent Node	New Node created after Add Node selected
None	Either a ROOT node or a Module Definition Node.
ROOT Node	PMA Node
PMA Node	Sub-PMA Node
Sub-PMA Node	Component Node
Component Node	Component Node
Redundancy Node	Component Node
Module Definition	Component Node

The Add Node function will automatically select the correct node to be placed if there can only be one placed. Otherwise, it will create a component node.

4.1.2 Add Redundancy

The Add Redundancy option will create a Redundancy node under the current node. There must be a current node for it to be added to. The Redundancy node can only be added to another Redundancy node, a Component node, a Module Definition node, or a Sub-PMA node.

4.1.3 Add Module

The Add Module option will add a Module node to the fault tree description. The function works differently, depending on the position of the cursor.

If there is cursor, that is no current module being displayed, the function will create a new Module Definition node.

If the cursor is on a Redundancy node, a Component node, a Module Definition node, or a Sub-PMA node, the function will add a branch to an existing Module Reference. It does this by allowing you to select the Module Definition from a given list box.

4.1.4 *Add Root*

This function will create the ROOT node of the fault tree description. It can only be created if no other nodes have been defined. Therefore it can only be used once, if that, during the creation of a description.

4.1.5 *Add Branch*

Using the Add Branch function you can create a link from one node to another existing node. The program will prompt you with a list box that contains a list of all the Module Definitions and the all the nodes in the current module. The Module Definitions appear first in the alphabetically sorted list, followed by the nodes, if any. A branch may only be added to a Redundancy node, a Component node, a Module Definition node, or a Sub-PMA node.

4.2 *File Menu*

The File Menu allows you to save and close modules; print the fault tree data; and exit the program. The file menu is shown below.

File Menu
Save Module
Close Module
Compile FTA Files
Print Module Tree
Print All Tree
Print Module Components
Print All Components
Exit

4.2.1 *Save Module*

The Save Module function will save the current module. Individual modules must be saved before moving to another module or exiting the program. The program will prompt you to save the module if you attempt to leave it without saving.

4.2.2 Close Module

This function will remove the current module from the fault tree editor. If it has changed, the program will prompt to save first. The Close Module function is used to save the current Module before creating a new module by clearing the current module out of memory, leaving a blank screen to start editing.

4.2.3 Compile FTA Files

This function will compile the current fault tree description files into a format useable by the GVAM suite of programs. The program will create two output files; a *.cmp component file, and a *.fta fault tree file. The name of the files will be the first word of the ROOT node description.

4.2.4 Print Module Tree

The Print Module Tree function will print out the graphical tree structure of the currently loaded module. The tree is printed to the data file named "print.tree" and is formatted for HPGL devices.

4.2.5 Print All Tree

The Print All Tree function is similar to the Print Module Tree function except it will print all the tree segments of the fault tree description. The tree is printed to the data file "print.tree" and is in HPGL format.

4.2.6 Print Module Components

This function will print a formatted list of the components in the current module. The data is printed to the data file "print.cmp" and is formatted for a 132 column line printer.

4.2.7 Print All Components

The function Print All Components will print a formatted list of all the components in the fault tree description to the data file "print.cmp". The data is formatted for a 132 column line printer.

4.2.8 Exit

The Exit function will leave the GRAFTED program. If the current module has been modified, the program will prompt with a warning and the option to save the changed module first.

4.3 Goto Menu

The Goto Menu allows you to move round the fault tree description by selecting key items of the nodes in the description. The Goto Menu is shown below.

Goto Menu
Goto Module
Goto Node
Goto Node ID
Goto Node Parent
Goto Last Module

Because of the multi-user features of the program, sometimes moving between modules will fail. This is because someone else is editing the module you wish to edit. When the function fails you will be returned to your last position.

4.3.1 Goto Module

The Goto Module function allows you to change position in the fault tree by selecting the name of a predefined module. The program will prompt you with a list box from which the module name can be selected. The module will not be opened if it is being edited by another user. The Goto Module function can also be selected by hitting the "Find" key on the keyboard, or by selecting the "Goto Module" icon in the menu area.

4.3.2 Goto Node

This function allows you to find a particular node in the fault tree and then position the cursor on it once the relevant module is loaded. The function will prompt you with a list box with all the fault tree components (excluding Module Definitions and Redundancies), from which the program will jump to the selected node. The function will fail if the module that contains the node is being edited by another user and hence cannot be loaded.

4.3.3 Goto Node ID

The Goto Node ID will search for a node by specifying its node identification number. The program will prompt for the Node ID, an integer below 20,000, and then attempt to find it. The function will fail if the component identifier does not exist, or the module that contains the node is being edited by another user and hence cannot be loaded.

4.3.4 *Goto Node Parent*

This function will prompt you with a list of the current module's parents and then jump to the selected node. This function will only work if the currently selected node is a Module Definition. If there is only one parent, the program will go to it immediately, otherwise you will be prompted with a list of the node's parents. Again, the function will fail if the module that was selected is being edited by another user and hence cannot be loaded.

4.3.5 *Goto Last Module*

The Goto Last Module function will return to the previous module that was being edited. It can also be invoked by hitting the "Ctrl-B" key. The program will remember the last 40 modules that have been edited. The function will fail if the last module is being edited by another user and hence cannot be loaded.

4.4 *Delete Menu*

The delete menu allows you to delete a node or a branch to a node. The delete menu is shown below.

Delete Menu
Delete Node
Delete Branch

4.4.1 *Delete Node*

The Delete Node option will attempt to delete the current node the cursor is positioned on. The option can also be executed by hitting the "Remove" key. A prompt to confirm the delete will first be given, then if the node can be deleted, it will be. A node cannot be deleted if any of the following conditions is true:

- the node is the ROOT node. The ROOT node can never be deleted,
- the node has more than one child and more than one parent,
- the node is a Module Definition and has nodes referencing it,
- the node is a Module Definition, PMA or Sub-PMA, and has children, or
- one of the GRAFTED data files is locked by another user.

If the delete fails for any reason, no changes will occur.

4.4.2 *Delete Branch*

The Delete Branch option will delete a branch from one node to another. The program will first prompt for the branch to be deleted. If there is only one branch going to the node, you will be asked to confirm deleting the node as well. Otherwise, the branch is removed.

4.5 *Zoom In/Zoom Out*

The Zoom In and Zoom Out icons change the size of the tree being displayed on the screen. The keyboard may also be used, pressing Ctrl-I to Zoom In, and Ctrl-O to Zoom Out. Zooming out will make more of the tree visible on the screen, hence making the tree features appear smaller. Zoom In will reverse the Zoom Out function. Zooming is set to operate at only three levels: 100%, 50%, and 25%.

4.6 *Refresh*

The Refresh Icon, also invoked by pressing Ctrl-R, will reset the screen if anything interrupts it. This is a common problem when the program is being run from a terminal.

5. *References*

[1] Tkalcevic F and Burman N M, *GRAFTED—Graphical Fault Tree Editor: A Fault Tree Description Program for Target Vulnerability/Survivability Analysis*, MRL–GD–0043, Materials Research Laboratory, Melbourne, Australia, 1992.

SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED

REPORT NO.	AR NO.	REPORT SECURITY CLASSIFICATION
MRL-GD-0057	AR-008-614	Unclassified

TITLE

GRAFTED - GRAphical Fault Tree EDitor: A fault tree description program for target vulnerability/survivability analysis - user manual

AUTHOR(S)
Michael E. Buckland, Mark Webster and
Frank J. Tkalcevic

CORPORATE AUTHOR
DSTO Materials Research Laboratory
PO Box 50
Ascot Vale Victoria 3032

REPORT DATE	TASK NO	SPONSOR
December, 1993	NAV 92/339	RAN

FILE NO.	REFERENCES	PAGES
G6/4/8-4501	1	27

CLASSIFICATION/LIMITATION REVIEW DATE	CLASSIFICATION/RELEASE AUTHORITY
	Chief, Ship Structures and Materials Division

SECONDARY DISTRIBUTION

Approved for public release

ANNOUNCEMENT

Announcement of this report is unlimited

KEYWORDS

Fault tree	Vulnerability	Survivability

ABSTRACT

A computer program GRAFTED, "GRAphical Fault Tree EDitor", has been written to simplify data entry and modification of component fault tree descriptions (FTD) used in military platform vulnerability/survivability analysis. GRAFTED uses a unique, graphical, screen based data entry procedure to define and display both individual system component parameters, and their hierarchical relationship in the overall system FTD. The generated component and system FTD output is in a format that is directly readable by the MRL version of the General Vulnerability Assessment Model, (GVAM), suite of computer programs.

Although GRAFTED was specifically designed to generate FTDs for GVAM, it could be easily modified to accommodate data input formats and FTD output for other assessment procedures that require user friendly data entry and graphical fault tree editing and visualisation.

This version of GRAFTED was written specifically for Tektronix UNIX workstations, and only runs on these machines.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

GRAFTED - GRAphical Fault Tree EDitor: A Fault Tree Description Program for Target Vulnerability/Survivability Analysis - User Manual

Michael E. Buckland, Mark Webster and Frank J. Tkalcevic

(MRL-GD-0057)

DISTRIBUTION LIST

Director, MRL - title page only
Chief, Ship Structures and Materials Division
Dr G. Johnston
Michael E. Buckland
Mark Webster
Frank J. Tkalcevic
MRL Information Services

Chief Defence Scientist (for CDS, FASSP, ASSCM) 1 copy only
Director (for Library), Aeronautical Research Laboratory
Head, Information Centre, Defence Intelligence Organisation
OIC Technical Reports Centre, Defence Central Library
Officer in Charge, Document Exchange Centre 8 copies
Air Force Scientific Adviser, Russell Offices
Navy Scientific Adviser
Scientific Adviser, Policy and Command
Senior Librarian, Main Library DSTOS
Librarian, DSD, Kingston ACT
Serials Section (M List), Deakin University Library, Deakin University, Geelong 3217
NAPOC QWG Engineer NBCD c/- DENGRS-A, HQ Engineer Centre, Liverpool Military Area, NSW 2174
ABCA, Russell Offices, Canberra ACT 2600 4 copies
Librarian, Australian Defence Force Academy
Head of Staff, British Defence Research and Supply Staff (Australia)
NASA Senior Scientific Representative in Australia
INSPEC: Acquisitions Section Institution of Electrical Engineers
Head Librarian, Australian Nuclear Science and Technology Organisation
Senior Librarian, Hargrave Library, Monash University
Library - Exchange Desk, National Institute of Standards and Technology, US
Exchange Section, British Library Document Supply Centre
Acquisition Unit (DSC-EO/GO), British Library, Boston Spa, Wetherby Yorkshire LS23 7BQ, England
Library, Chemical Abstracts Reference Service
Engineering Societies Library, US
Documents Librarian, The Center for Research Libraries, US
Army Scientific Adviser, Russell Offices - data sheet only
Director General Force Development (Land) - data sheet only
DASD, APW2-1-OA2, Anzac Park West, Canberra ACT - data sheet only
SO (Science), HQ 1 Division, Milpo, Enoggera, Qld 4057 - data sheet only
Librarian - MRL Sydney - data sheet only
Counsellor, Defence Science, Embassy of Australia - data sheet only
Counsellor, Defence Science, Australian High Commission - data sheet only
Scientific Adviser to DSTC Malaysia, c/- Defence Adviser - data sheet only
Scientific Adviser to MRDC Thailand, c/- Defence Attache - data sheet only